IFI TECHNICAL REPORTS

Institute of Computer Science,
Clausthal University of Technology

IfI-05-14

Clausthal-Zellerfeld 2005

On the generalization ability of prototype-based classifiers with local relevance determination

Barbara Hammer[1], Frank-Michael Schleif[2], Thomas Villmann[3]

1 - Institute of Computer Science, Clausthal University of Technology, Germany,
2 - Bruker Daltonics, Leipzig, Germany,
3 - Clinic for Psychotherapy, Universität Leipzig, Germany

Abstract

We extend a recent variant of the prototype-based classifier learning vector quantization to a scheme which locally adapts relevance terms during learning. We derive explicit dimensionality-independent large-margin generalization bounds for this classifier and show that the method can be seen as margin maximizer.

1 Introduction

Prototype-based classifiers constitute simple though powerful learning models with very intuitive classification behavior since the prototypes are located at representative regions of the same space as the training data. There exist numerous methods for prototype adaptation including unsupervised models such as self-organizing maps or neural gas [25, 28] and supervised methods such as learning vector quantization (LVQ) and variants thereof [19, 25, 26, 36]. LVQ is particularly interesting due to the simplicity of the learning algorithm, and it has successfully been applied in various areas including satellite image processing, time series processing, robotics, linguistics, handwritten digit recognition, bioinformatics, etc. [18, 25, 38, 43].

LVQ is based on a heuristic and divergence or instable behavior can be observed frequently. Therefore, several extensions have been proposed including methods such as LVQ2.1, LVQIII, OLVQ. Only few methods, however, are accompanied by an objective function [19, 32, 36] and an exact mathematical analysis of the behavior of LVQ-type learning algorithms and their generalization curves has just started [3]. Interestingly it can be shown that LVQ-type classifiers based on the euclidian metric can be interpreted as large margin classifiers for which dimensionality independent generalization bounds exist [7]. However, the influence of different training algorithms on the size of the margin and the generalization behavior is often unclear. Generalized relevance LVQ (GRLVQ) as introduced in [19] constitutes one notable exception. It directly optimizes an objective function which contains a term characterizing the margin.

Prototype-based algorithms heavily depend on the metric which is used for comparison, usually the euclidian metric. This makes them unsuitable for high dimensional noisy data or heterogeneous scaling of the dimensions. Since a proper scaling or preprocessing of data is often not available, methods which automatically determine optimum metric parameters based on additional information are particularly interesting. This includes methods for input selection [10, 11, 31, 39], metric adaptation in unsupervised learning [9, 13, 23, 24], and supervised methods based on a global cost function [16, 19, 42].

We are interested in methods which adapt the metric locally according to the given task such that an optimum scaling is found at each point of the data space. It is well known that local or class-wise adaptation of parameters can play a major role for the classification accuracy and flexibility in comparison to global scalings, see e.g. the relation of linear discriminant analysis (with global parameters) to quadratic discriminant analysis (with class-wise parameters). Local metric schemes have been introduced e.g. for unsupervised fuzzy classifier [9, 13]. For GRLVQ networks, the generalization of the update rules to local parameters is straightforward, as we will see in this paper. However, it is not clear whether the good generalization bounds as developed in [17] still hold, since the metric is changed during training using more degrees of freedom. We will show in this paper that large-margin generalization bounds can also be derived for local GRLVQ-type networks. The bounds hold for the locally adaptive scaled euclidian metric and any version which can be interpreted as a kernelization thereof, as explained e.g. in [16, 34].

2 Prototype-based classification

From a mathematical point of view, we are interested in general classification tasks. Data $X = \{x^i \in \mathbb{R}^n \,|\, i = 1, \ldots, m\}$, whereby the input vectors x^i are characterized by n features, are to be classified into C given classes. Components of a vector $x \in \mathbb{R}^n$ are referred to by subscripts, i.e., $x = (x_1, \ldots, x_n)$. Prototype-based classifiers constitute a particularly intuitive way of classification by means of typical locations of known class allocation which characterize local regions of the data space. Every class c is represented by a set $W(c)$ of weight vectors (prototypes) in $\mathbb{R}^n$. Weight vectors are denoted by w^r and their respective class label is referred to by c_r. A new signal $x \in \mathbb{R}^n$ is classified by the winner-takes-all rule of the classifier, i.e.

$$x \mapsto c(x) = c_r \text{ such that } d(x, w^r) \text{ is minimum}. \tag{1}$$

Thereby, $d(x, w^r)$ is chosen as the squared Euclidean distance

$$d(x, w^r) = \|x - w^r\|^2 = \sum_{i=1}^{n} (x_i - w_i^r)^2 \tag{2}$$

of the data point x to the prototype w^r. The respective closest prototype w^r is called winner or best matching unit for x. The subset

$$\Omega_r = \{x^i \in X \mid d(x^i, w^r) \text{ is minimum}\}$$

is called receptive field of neuron w^r. Thus, data point x^i is mapped to the class $c(x^i)$.

2.1 Learning vector quantization schemes

Usually, one is interested in finding a prototype-based classifier which matches a given training set and its underlying regularity as accurately as possible. A training set consists of a collection of data points together with their known class allocations $\{(x^i, y_i) \in \mathbb{R}^n \times \{1, \ldots, C\} \mid i = 1, \ldots, m\}$. Training aims at minimizing the classification error on the given training set. I.e., prototype locations have to be found such that the difference between the set of points belonging to the cth class, $\{x^i \in X \mid y_i = c\}$ and the receptive fields of the corresponding prototypes, $\bigcup_{w^r \in W(c)} \Omega_r$, is minimized by the adaptation process.

Learning vector quantization (LVQ) as proposed by Kohonen [26] constitutes a popular and simple learning algorithm which forms the base for several extensions and alternatives. The LVQ learning rule consists in heuristically motivated Hebbian learning: iteratively, a data point x^i is randomly chosen from the training set and the respective winner w^r is adapted in the following way

$$\triangle w^r = \begin{cases} \epsilon \cdot (x^i - w^r) & \text{if } c^r = c(x^i) \\ -\epsilon \cdot (x^i - w^r) & \text{otherwise}\,. \end{cases}$$

$\epsilon \in (0, 1)$ is an appropriate learning rate. As explained in [32], this update can be interpreted as a stochastic gradient descent on the cost function

$$\text{Cost}_{\text{LVQ}} = \sum_{x^i \in X} f_{\text{LVQ}}(d_{r_+}, d_{r_-})\,.$$

d_{r_+} denotes the squared Euclidean distance of x^i to the closest prototype w^{r+} labeled with $c_{r_+} = y_i$, and d_{r_-} denotes the squared Euclidean distance to the closest prototype w^{r-} labeled with a label c_{r_-} different from y_i. For standard LVQ, the function is

$$f_{\text{LVQ}}(d_{r_+}, d_{r_-}) = \begin{cases} d_{r_+} & \text{if } d_{r_+} \leq d_{r_-} \\ -d_{r_-} & \text{otherwise} \end{cases}$$

Obviously, this cost function is highly discontinuous, and instabilities arise for overlapping data distributions.

Various alternatives have been proposed which substitute the training rule of LVQ by alternatives in order to achieve more stable training in case of overlapping classes or noisy data. Kohonen's LVQ2.1 optimizes the cost function which is obtained by setting

in the above sum $f_{\mathrm{LVQ2.1}}(d_{r_+}, d_{r_-}) = I_w(d_{r_+} - d_{r_-})$, whereby I_w yields the identity inside a window where LVQ2.1 adaptation takes place, and I_w vanishes outside. Still this choice might produce an instable dynamic, and the window where adaptation takes place must be chosen carefully. Generalized LVQ (GLVQ) has been proposed by Sato and Yamada as a stable alternative to LVQ2.1 derived from a more appropriate cost function [32]. The respective cost function can be obtained by setting

$$f_{\mathrm{GLVQ}}(d_{r_+}, d_{r_-}) = \mathrm{sgd}\left(\frac{d_{r_+} - d_{r_-}}{d_{r_+} + d_{r_-}}\right)$$

whereby $\mathrm{sgd}(x) = (1 + \exp(-x))^{-1}$ denotes the logistic function. As discussed in [33], the additional scaling factors avoid numerical instabilities and divergent behavior. The update rule can be achieved by taking the derivatives [16]

$$\triangle w^{r_+} = \epsilon^+ \cdot \mathrm{sgd}'_{\mu(x^i)} \cdot \xi^+ \cdot 2 \cdot (x^i - w^{r_+})$$

and

$$\triangle w^{r_-} = -\epsilon^- \cdot \mathrm{sgd}'_{\mu(x^i)} \cdot \xi^- \cdot 2 \cdot (x^i - w^{r_-})$$

where ϵ^+ and $\epsilon^- \in (0, 1)$ are the learning rates, the logistic function is evaluated at position $\mu(x^i) = (d_{r_+} - d_{r_-})/(d_{r_+} + d_{r_-})$, and

$$\xi^+ = \frac{2 \cdot d_{r_-}}{(d_{r_+} + d_{r_-})^2} \quad \text{and} \quad \xi^- = \frac{2 \cdot d_{r_+}}{(d_{r_+} + d_{r_-})^2}$$

denote the derivatives of $f_{\mathrm{GLVQ}}(d_{r_+}, d_{r_-})$ with respect to d_{r_+} and d_{r_-}, respectively.

This procedure still has the drawback that it is very sensitive to initialization of prototypes because of the multiple optima of the cost function. This can be widely avoided by integrating neighborhood cooperation of the prototypes into the learning scheme. Neural gas (NG) constitutes a popular and robust unsupervised vector quantizer based on a data optimum neighborhood structure [28, 29]. The cost function of GLVQ allows to integrate the neighborhood cooperation scheme of NG into the learning vector quantization, yielding supervised neural gas (SNG). The global cost function becomes

$$E_{\mathrm{SNG}} = \sum_{x^i \in X} \sum_{w^r \in W(y_i)} \frac{h_\gamma(r, x^i, W(y_i)) \cdot f_{\mathrm{SNG}}(d_r, d_{r_-})}{C(\gamma, K_{y_i})}$$

whereby

$$f_{\mathrm{SNG}}(d_r, d_{r_-}) = f_{\mathrm{GLVQ}}(d_r, d_{r_-}) = \mathrm{sgd}\left(\frac{d_r - d_{r_-}}{d_r + d_{r_-}}\right)$$

and d_r denotes the squared Euclidian distance of x^i to w^r.

$$h_\gamma(r, x^i, W(y^i)) = \exp\left(-\frac{k_r(x^i, W(y_i))}{\gamma}\right)$$

denotes the degree of neighborhood cooperativity, $k_r(x^i, W(y_i))$ yielding the number of prototypes w^p in $W(y_i)$ for which $d_p \leq d_r$ is valid, i.e. the rank of w_r. $C(\gamma, K_{y_i})$ is a normalization constant depending on the neighborhood range γ and cardinality K_{y_i} of $W(y_i)$. w^{r-} denotes the closest prototype not in $W(y_i)$. Here *all* prototypes of a specific class are adapted towards the given data point, preventing neurons from being idle or repelled from their class. A superposition of the NG and GLVQ dynamics within these update rules assures a stable and robust convergence of the algorithm towards good optima: the NG-dynamics aims at spreading all prototypes with a specific class label faithfully among the respective data. The simultaneous GLVQ dynamics makes sure that those class borders are found which yield a good classification. Note that vanishing neighborhood cooperativity $\gamma \to 0$ yields the original cost function of GLVQ.

As beforehand, the update formulas for the prototypes can be obtained taking the derivative [16]. For each x^i, all prototypes $w^r \in W(y_i)$ are adapted by

$$\triangle w^r = \epsilon^+ \cdot \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot \xi_r^+ \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{y_i})} \cdot 2 \cdot (x^i - w^r)$$

and the closest wrong prototype is adapted by

$$\triangle w^{r-} = -\epsilon^- \cdot \sum_{w^r \in W(y_i)} \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot \xi_r^- \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{y_i})} \cdot 2 \cdot (x^i - w^{r-})$$

whereby ϵ^+ and $\epsilon^- \in (0, 1)$ are learning rates and the logistic function is evaluated at position

$$\mu^r(x^i) = \frac{d_r - d_{r_-}}{d_r + d_{r_-}} \,.$$

The terms ξ are again obtained as derivative of f_{SNG} as

$$\xi_r^+ = \frac{2 \cdot d_{r_-}}{(d_r + d_{r_-})^2} \quad \text{and} \quad \xi_r^- = \frac{2 \cdot d_r}{(d_r + d_{r_-})^2} \,.$$

As shown in [16], these derivatives also exist for an underlying continuous data distribution. Note that the original updates of GLVQ are recovered if $\gamma \to 0$. For positive neighborhood cooperation, all correct prototypes are adapted according to a given data point such that also neurons outside their class become active. Eventually, neurons become spread among the data points of their respective class. Since all prototypes have thereby a repelling function on the closest incorrect prototype, it is advisable to choose ϵ^- one magnitude smaller than ϵ^+.

2.2 Metric adaptation

Prototype-based classifiers crucially depend on the metrics. If the Euclidean metric is chosen, it is implicitly assumed that all input dimensions have the same relevance for

the classification since each dimension contributes equally to the computed distances. This causes problems if high dimensional data, data affected by noise, or data descriptions with different but possibly unknown relevance are considered. Noise or irrelevant dimensions may disrupt the information contained in the relevant attributes. This effect accumulates for high dimensionality, and it is made worse by the curse of dimensionality. Thus, either extensive preprocessing and feature extraction prior to training is necessary – but time consuming – or a choice of a different, problem adapted metric is advisable. Since an appropriate metric is usually not clear prior to learning, learning metrics which are automatically adapted during training according to the information contained in the data are particularly interesting.

In general, the Euclidean metric (2) can be substituted by a different choice which might include adaptive parameters λ. Since GLVQ and SNG are formulated as general cost minimization algorithms, any differentiable similarity measure can be integrated into its cost functions yielding update rules for the prototypes where the Hebbian terms $(x^i - w)$ are substituted by the derivative of the respective similarity measure with respect to the prototype w, as demonstrated in [16]. The same optimization mechanism can be used to adapt metric parameters during training. Then the prototype update is accompanied by a simultaneous adaptation of the metric parameters given by the derivative of the cost function with respect to these metric parameters. Several alternatives to the squared Euclidean metric such as metrics better adapted for time series data have been proposed [16]. One simple extension of the squared Euclidean metric proved particularly efficient and powerful, which has the additional benefits that it allows a natural interpretation of the results, and it can be accompanied by theoretical guarantees for its good generalization ability: the Euclidean metric enhanced by adaptive relevance terms for the input dimensions which we introduce now.

We substitute the squared Euclidean metric (2) by the term

$$d^\lambda(x, w^r) = \|x - w^r\|^2_\lambda = \sum_{i=1}^{n} \lambda_i \cdot (x_i - w^r_i)^2$$

whereby $\lambda = (\lambda_1, \ldots, \lambda_n)$ with $\lambda_i \geq 0$ contains nonnegative relevance terms with the constraint $\sum_{i=1}^{n} \lambda_i = 1$. The GLVQ and SNG update rules for the prototypes remain widely the same except for an extension of the metric by relevance terms and additional factors λ_i within the Hebb term. For the relevance terms, the update becomes

$$\triangle\lambda_l = -\epsilon_\lambda \cdot \sum_{w^r \in W(y_i)} \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{c_\mathbf{v}})} \cdot \left(\xi^+_r \cdot (w^r_l - x^i_l)^2 - \xi^-_r \cdot (w^{r^-}_l - x^i_l)^2\right)$$

for relevance determination in SNG, supervised relevance neural gas (SRNG), and

$$\triangle\lambda_l = -\epsilon_\lambda \cdot \mathrm{sgd}'|_{\mu^r(x^i)} \cdot \left(\xi^+ \cdot (w^{r+}_l - x^i_l)^2 - \xi^- \cdot (w^{r^-}_l - x^i_l)^2\right)$$

for generalized relevance LVQ (GRLVQ). As discussed in [16], this adaptation scheme can be interpreted as intuitive Hebbian learning for the relevance terms. Here $\epsilon_\lambda \in$

$(0, 1)$ is the learning rate for the relevance terms. The constraints $\lambda_l \geq 0$ and $\sum_l \lambda_l = 1$ are enforced by an explicit normalization of the relevance terms after each adaptation step.

Note that a relevance profile is automatically determined during training which allows an interpretation of the results. If dimensions are scaled equally at the beginning of training, high relevance values during training indicate that the dimension has a large contribution to the classifier whereas small values indicate dimensions which hardly influence the classification result. Thus, apart from an improved classification accuracy, relevance adaptation allows to gain insight into the behavior of the model and to determine the importance of the input dimensions for the classification task. As shown in the article [15], this additional information can be used to directly extract approximate decision trees from the classifier, i.e., an explicit approximate description of the classification by symbolic rules.

2.3 Generalization ability

Alternative cost functions and adaptation schemes for LVQ-type classification have been proposed and metric adaptation and relevance determination for alternative, in particular unsupervised models have been derived in a variety of articles, see e.g. [2, 5, 8, 9, 10, 11, 13, 21, 23, 24, 30, 31, 36, 39, 42]. In addition, different metric choices might prove beneficial, in particular if complex, possibly non-vectorial data are to be dealt with [8, 12, 16, 18, 27]. The question arises which choice of a cost function and which metric is in general best suited. GRLVQ and SRNG have several benefits which make them attractive in a number of classification tasks ranging from applications for time series prediction, bioinformatics, up to satellite image processing [16, 18, 38, 43].

Assume there is given a prototype-based classifier which maps data to classes according to the winner-takes-all rule (1). It has been shown that the term

$$(\|x^i - w^{r-}\| - \|x^i - w^{r+}\|)/2$$

constitutes the so-called hypothesis margin of such a prototype-based classifier [7]. The hypothesis margin refers to the distance in an appropriate norm, which the classifier can alter without changing the classification. Generalization bounds which depend on this hypothesis margin have been derived in [7]. Note that LVQ2.1, GLVQ, and SNG express this margin in terms of their cost functions, hence, they can be interpreted as margin optimization learning algorithms comparable to support vector machines, which aim at directly optimizing the margin, i.e. generalization bound of the classifier during training [6, 40, 41]. Thus the chosen cost function combines stability and a good classification accuracy with robustness with respect to the generalization ability.

The notion of learning metrics introduces further degrees of freedom into the classifier. It is obvious that an adaptive metric can be crucial for a good classification accuracy in the same way as the design of a kernel for a support vector machine constitutes an essential part of the model design. Since this part is often time consuming, an

automatic adaptation of this part according to the given data is highly desirable. However, an arbitrary adaptation of the metric or kernel might disrupt the generalization ability of the classifier and generalization bounds do no longer hold [4]. It has recently been shown that GRLVQ or SRNG-type networks with adaptive diagonal metrics can also be interpreted as large margin optimization algorithms, and explicit dimensionality independent generalization bounds which only depend on the quantity

$$\|x^i - w^{r-}\|^2_\lambda - \|x^i - w^{r+}\|^2_\lambda$$

have been derived [17]. Note that these bounds are valid for adaptive relevance terms λ. Thus, SRNG and GRLVQ retain the generalization ability of LVQ networks and the large margin optimization property of SNG and GLVQ, whereby larger flexibility because of the adaptive metric and the possibility to gain further information by means of the relevance profile is achieved.

It should be mentioned that this fact directly transfers to all metrics which can be interpreted as a kernelized version of original GRLVQ and SRNG. Thereby, a kernel function consist of a mapping $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that some Hilbert space X and a function $\Phi : \mathbb{R}^n \to X$ can be found with

$$k(x, y) = \Phi(x)^t \Phi(y)$$

i.e. k can be interpreted as scalar product in some high dimensional (possibly infinite dimensional) space. The most prominent application of kernels within machine learning can be found in the context of SVMs [6]. However, based on the success of SVM, kernelization of various alternative machine learning tools such as principal and independent component analysis became popular [35]. If the chosen kernel is fixed, results from statistical learning theory such as bounds on the generalization error can be transferred directly from the basic version of the learning algorithm to the kernelized one. At the same time, appropriate nonlinear kernels often considerably expand the capacity of the original method, yielding universal approximators in the case of SVM, for example [14, 37]. Thereby, the possibly high dimensional mapping Φ need not be computed explicitely such that the computational effort can be reduced. The fact whether a function constitutes a kernel can be tested using e.g. Mercer's theorem [35]. Popular kernels include, for example, the polynomial kernel, the Gaussian kernel, or kernels specifically designed for complex data structures such as strings [20, 22].

In our case, we are interested in a general similarity measures d included in our cost function such that some $\Phi : \mathbb{R}^n \to X$ exists with

$$d(x, y) = \|\Phi(x) - \Phi(y)\|^2_\lambda$$

whereby $\|\cdot\|$ denotes the metric in the Hilbert space X. If this holds, we can interprete the cost function E_{SRNG} as cost function of SRNG in some (possibly) high dimensional Hilbert space, whereby the generalization ability of the classifier only depends on the margin of the classifier. It is well known that such Φ can be found for a more

general class of functions than Mercer kernels: one sufficient condition for an equality $d(x,y) = \|\Phi(x) - \Phi(y)\|$ is, for example, that d constitutes a real-valued symmetric functions d with $d(x,x) = 0$ for all x such that $-d$ is conditionally positive definite, i.e. for all $N \in \mathbb{N}$, $c_1, \ldots, c_N \in \mathbb{R}$ with $\sum_i c_i = 0$ and $x^1, \ldots, x^N \in \mathbb{R}^n$ the inequality $\sum_{i,j} c_i c_j \cdot (-1) \cdot d(x^i, x^j) \geq 0$ holds [34]. As an example, functions of the form $\|\mathbf{x} - \mathbf{y}\|^\beta$ for an arbitrary metric $\|\cdot\|$ and $\beta \in (0, 2]$ fulfill these properties.

3 Local relevance adaptation

In decision making, in particular medical classifications, the relevance of the input features usually depends on the considered classes and the region of the data space. An indicative feature for a particular disease (A) might be entirely unrelated to a different disease. Thus, this feature need only be taken into account if disease (A) is considered. Otherwise, it might disrupt the classification since it only contributes noise to alternative considerations. This situation can also be observed in hierarchical decision schemes, where a particular feature might only be relevant at the first level whereas different features determine the classification within deeper decision levels. To take these considerations into account, we extend prototype-based classifiers by *local relevance* factors connected to the specific prototypes and hence the specific regions of the data spaces.

3.1 Local GRLVQ and SRNG

Assume, as beforehand, a set of training data $\{(x^i, y_i) \in \mathbb{R}^n \times \{1, \ldots, C\} \mid i = 1, \ldots, m\}$ is given and prototypes w^r with class label c_r are fixed. Here we introduce relevance factors

$$\lambda^r = (\lambda_1^r, \ldots, \lambda_n^r)$$

with the constraint $\lambda_i^j \geq 0$ and $\sum_i \lambda_i^j = 1$ attached to prototype r. Thus, the relevance factors are assigned to a specific prototype and the respective local region of the data space. They can be adapted independently for each local region of the data space.

$$d_r^{\lambda^r}(x, w^r) = \|x - w^r\|_{\lambda^r}^2 = \sum_{i=1}^{n} \lambda_i^r (x_i - w_i^r)^2$$

denotes the local metric used by prototype r. Classification is performed extending the winner takes all rule to this situation

$$x \mapsto c(x) = c_r \text{ such that } \|x - w^r\|_{\lambda^r}^2 \text{ is minimum}\,. \tag{3}$$

Note that now, the receptive fields of prototypes need no longer be convex since no global metric is used for classification. They account for the local characteristic of the

data space and they take the local relevance profile into account. In particular, they allow more complex decision shapes compared to global metric parameters.

Training can be achieved by taking the derivative of the extended cost functions

$$\text{Cost}_{LGRLVQ} = \sum_{x^i \in X} \text{sgd}\left(\frac{d_{r_+}^{\lambda^{r_+}} - d_{r_-}^{\lambda^{r_-}}}{d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}}}\right)$$

and

$$\text{Cost}_{LSRNG} = \sum_{x^i \in X} \sum_{w^r \in W(y_i)} \frac{h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{y_i})} \cdot \text{sgd}\left(\frac{d_r^{\lambda^r} - d_{r_-}^{\lambda^{r_-}}}{d_r^{\lambda^r} + d_{r_-}^{\lambda^{r_-}}}\right)$$

where $h_\gamma(r, x^i, W(y_i)) = \exp(-k_r(x^i, W(y_i))/\gamma)$ now measures the neighborhood range with respect to the number of prototypes $w^p \in W(y_i)$ for which

$$d_p^{\lambda^p}(x, w^p) \leq d_r^{\lambda^r}(x, w^r)$$

is valid.

The updates for the prototypes and local relevance terms are achieved taking the derivatives as beforehand. Local GRLVQ (LGRLVQ) is given by the rules

$$\triangle w^{r_+} = \epsilon^+ \cdot \text{sgd}'_{\mu(x^i)} \cdot \xi^+ \cdot 2 \cdot \Lambda^{r_+} \cdot (x^i - w^{r_+})$$

for the closest correct prototype and

$$\triangle w^{r_-} = -\epsilon^- \cdot \text{sgd}'_{\mu(x^i)} \cdot \xi^- \cdot 2 \cdot \Lambda^{r_-} \cdot (x^i - w^{r_-})$$

for the closest wrong prototype where the logistic function is evaluated at position $\mu(x^i) = (d_{r_+}^{\lambda^{r_+}} - d_{r_-}^{\lambda^{r_-}})/(d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}})$, and

$$\xi^+ = \frac{2 \cdot d_{r_-}^{\lambda^{r_-}}}{(d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}})^2} \quad \text{and} \quad \xi^- = \frac{2 \cdot d_{r_+}^{\lambda^{r_+}}}{(d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}})^2}\,.$$

Λ^r denotes the diagonal matrix with entries λ_i^r. The relevance terms are adapted by

$$\triangle \lambda_l^{r_+} = -\epsilon_\lambda \cdot \text{sgd}'|_{\mu^r(x^i)} \cdot \xi^+ \cdot (w_l^{r_+} - x_l^i)^2$$

and

$$\triangle \lambda_l^{r_-} = -\epsilon_\lambda \cdot \text{sgd}'|_{\mu^r(x^i)} \cdot \left(-\xi^- \cdot (w_l^{r_-} - x_l^i)^2\right)$$

For local SRNG (LSRNG) we achieve

$$\triangle w^r = \epsilon^+ \cdot \frac{\text{sgd}'|_{\mu^r(x^i)} \cdot \xi_r^+ \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{y_i})} \cdot 2 \cdot \Lambda^r \cdot (x^i - w^r)$$

for all correct prototypes, and the closest wrong prototype is adapted by

$$\triangle w^{r-} = -\epsilon^- \sum_{w^r \in W(y_i)} \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot \xi_r^- \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{y_i})} \cdot 2 \cdot \Lambda^{r-} \cdot (x^i - w^{r-})$$

whereby the logistic function is evaluated at position

$$\mu^r(x^i) = \frac{d_r^{\lambda^r} - d_{r-}^{\lambda^{r-}}}{d_r^{\lambda^r} + d_{r-}^{\lambda^{r-}}} \,.$$

The terms ξ are obtained as

$$\xi_r^+ = \frac{2 \cdot d_{r-}^{\lambda^{r-}}}{(d_r^{\lambda^r} + d_{r-}^{\lambda^{r-}})^2} \quad \text{and} \quad \xi_r^- = \frac{2 \cdot d_r^{\lambda^r}}{(d_r^{\lambda^r} + d_{r-}^{\lambda^{r-}})^2} \,.$$

Relevance terms are adapted by

$$\triangle \lambda_l^r = -\epsilon_\lambda \cdot \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{c_\mathbf{v}})} \cdot \xi_r^+ \cdot (w_l^r - x_l^i)^2$$

for all w^r of the correct class and

$$\triangle \lambda_l^{r-} = -\epsilon_\lambda \sum_{w^r \in W(y_i)} \frac{\mathrm{sgd}'|_{\mu^r(x^i)} \cdot h_\gamma(r, x^i, W(y_i))}{C(\gamma, K_{c_\mathbf{v}})} \cdot \left(-\xi_r^- \cdot (w_l^{r-} - x_l^i)^2\right) \,.$$

In both cases, a normalization for every λ^r is added after each adaptation. Note that these learning rules contain the standard Hebb terms in a local version for the parameters λ^r accompanied by additional factors which cause a better stability of the algorithms.

3.2 Generalization ability

We have introduced additional parameters of the classifier, such that bounds on the generalization ability of the simpler model do no longer hold for this extended and more powerful setting. The aim of this section is to derive large margin generalization bounds also for this more general case such that a proof for the good generalization capacity of this more flexible model becomes available. Thereby, we derive bounds for general function classes given by the winner takes all rule with adaptive local metric as defined in equation (3). Thus the error bounds hold for every classifier no matter how training takes place. In addition, we show that the denominator of the cost function function of local GRLVQ characterizes the margin and directly influences the generalization bound. Thus, LGRLVQ can be interpreted as large margin algorithm just as GRLVQ.

Generally speaking, the generalization ability of a classifier refers to the comparison of the training error with the expected error for new data. There are various ways

to formalize and prove the generalization ability of classifiers, such as the popular VC-theory [41] or recent argumentation based on Rademacher and Gaussian complexity [1]. Here, we consider the situation of binary classification problems, i.e. only two classes are given, and we assume the classes are labeled 1 and -1. Assume an (unknown) probability measure P is given on $\mathbb{R}^n \times \{-1, 1\}$. Training samples (x^i, y_i) are drawn independently and identically distributed (i.i.d. for short) from $\mathbb{R}^n \times \{-1, 1\}$. P^m refers to the product of P if m examples (x^1, y_1), $\ldots$, (x^m, y_m) are chosen. The unknown regularity shall be learned by a LGRLVQ-network or some other prototype-based classifier with adaptive local diagonal metric. The classifier is characterized by its set of prototypes $w^1, \ldots, w^p$ in $\mathbb{R}^n$ (p denoting the number of prototypes) and the respective relevance terms $\lambda^1, \ldots, \lambda^p$ which describe the local weighted metrics. The function computed by the classifier is given by the winner-takes-all rule defined in (3). Denote by

$$\mathcal{F} = \{f : \mathbb{R}^n \to \{-1, 1\} \mid f \text{ is given by (3) depending on } w^1, \ldots, w^p, \lambda^1, \ldots, \lambda^p \in \mathbb{R}^n\}$$

the class of functions which can be computed by such a network. The goal of learning is to find a function $f \in \mathcal{F}$ for which the probability

$$E_P(f) := P(y \neq f(x))$$

is minimum. Since the underlying regularity P is not known and only examples (x^i, y_i) are available for characterizing this regularity, training tries to minimize the empirical training error

$$\hat{E}_m(f) := \sum_{i=1}^{m} 1_{y_i \neq f(x^i)}/m$$

whereby $1_{y_i \neq f(x^i)}$ indicates whether x^i is mapped to the desired class y_i or not. Generalization means that $\hat{E}_m(f)$ is representative for $E(f)$ with high probability if the examples are chosen according to P^m such that optimization of the empirical training error will eventually approximate the underlying regularity.

Due to the chosen cost function, LGRLVQ minimizes the training error and, in addition, also optimizes the margin of the classifier during training. Given a point x with desired output y, we define the margin as the value

$$M_f(x, y) := -d_{r_+}^{\lambda^{r+}} + d_{r_-}^{\lambda^{r-}}$$

whereby $d_{r_+}^{\lambda^{r+}}$ refers to the squared weighted distance of the closest prototype of the same class as x, and $d_{r_-}^{\lambda^{r-}}$ refers to the squared weighted distance of the closest prototype labeled with a different class from x. x is classified incorrectly iff $M_f(x, y)$ is negative. Otherwise, x is classified correctly with 'security' margin $M_f(x, y)$. Due to the choice of the cost function of LGRLVQ which involves this term within the denominator, LGRLVQ aims at maximizing this margin. Following the approach [1] we define

the loss function

$$L : \mathbb{R} \to \mathbb{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

for fixed $\rho > 0$. The term

$$\hat{E}_m^L(f) := \sum_{i=1}^{m} L(M_f(x^i, y_i))/m$$

accumulates the number of errors made by f and, in addition, punishes all correctly classified points, if their margin is smaller than ρ.

We will now show that this modified empirical error, which also includes the margin, is representative for the true error with high probability, whereby a bound which is independent of the dimensionality of the input space is obtained. We assume that the support of the probability measure P is bounded, i.e. that for all data points x the inequality

$$\|x\| \leq B$$

holds for some $B > 0$, $\|\cdot\|$ denoting the standard Euclidean metric. In addition, all prototypes w are restricted by

$$\|w\| \leq B\,.$$

According to [1](Theorem 7) we can estimate for all $f \in \mathcal{F}$ with probability at least $1 - \delta/2$

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{2K}{\rho} \cdot G_m(\mathcal{F}) + \sqrt{\frac{\ln(4/\delta)}{2m}}$$

whereby K is a universal positive constant and $G_m(\mathcal{F})$ is the so-called Gaussian complexity of the considered function class which we now define. The empirical Gaussian complexity is given by

$$\hat{G}_m(\mathcal{F}) = E_{g_1,\ldots,g_m}\left(\sup_{f\in\mathcal{F}}\left|\frac{2}{m}\sum_{i=1}^{m} g_i \cdot f(x^i)\right|\right)$$

for which expectation is taken with respect to independent Gaussian variables $g_1, \ldots, g_m$ with zero mean and unit variance. The Gaussian complexity is the expectation over the i.i.d. points x^i according to the marginal distribution induced by P: $G_m(\mathcal{F}) = E_{x^1,\ldots,x^m}\hat{G}_m(\mathcal{F})$. Both complexities measure the richness of the function class $\mathcal{F}$ and constitute convenient alternatives to the standard VC-dimension which can also be estimated for prototype-based classifiers.

The classification given by the winner-takes-all rule (3) can be reformulated as fixed Boolean formula over terms of the form $d_i^{\lambda^i} - d_j^{\lambda^j}$ with $d_i^{\lambda^i}$ and $d_j^{\lambda^j}$ constituting the weighted squared Euclidean distance of a given input x to two prototypes w^i and

w^j with different class labels. Note that the number of such terms is upper bounded by $p \cdot (p-1)/2$ since p prototypes are available within the classifier. According to [1](Theorem 16) we find

$$G_m(\mathcal{F}) \leq p \cdot (p-1) \cdot G_m(\mathcal{F}_{ij})$$

whereby $\mathcal{F}_{ij}$ denotes the restricted class of classifiers which can be implemented with only two prototypes w^i and w^j with different class label. Define by Λ^i the diagonal matrix with entries λ^i_j. For fixed i and j, we find

$$\begin{array}{ll} & d_i^{\lambda^i} - d_j^{\lambda^j} \leq 0 \\ \iff & (x-w^i)^t \cdot \Lambda^i \cdot (x-w^i) - (x-w^j)^t \cdot \Lambda^j \cdot (x-w^j) \leq 0 \\ \iff & x^t \cdot \Lambda^i \cdot x - x^t \cdot \Lambda^j \cdot x \\ & - 2 \cdot (\Lambda^i \cdot w^i - \Lambda^j \cdot w^j)^t x + (w^i)^t \cdot \Lambda^i \cdot w^i - (w^j)^t \cdot \Lambda^j \cdot w^j \leq 0 \end{array}$$

Hence, every function from $\mathcal{F}_{ij}$ can be written as the sum of a function from the set $\mathcal{F}_i = \{x \mapsto x^t \cdot \Lambda^i \cdot x\}$, a function from the set $-\mathcal{F}_j$, and a function implemented by a simple perceptron, i.e. linear classifier. According to [1](Theorem 12), it holds $G_m(c \cdot \mathcal{F}) = c \cdot G_m(\mathcal{F})$ and $G_m(\sum_i \mathcal{F}_i) \leq \ln m \sum_i G_m(\mathcal{F}_i)$. Thus it is sufficient to independently estimate the Gaussian complexity of linear and quadratic functions of this form.

For linear functions, the estimation follows immediately: since $\|x\| \leq B$, the length of inputs to the linear classifier can be restricted by $B+1$ (including the bias term). Since all prototypes w are restricted by $\|w\| \leq B$ and the relevance terms add up to 1, the size of the weights of the linear classifier is restricted by $4B + 2B^2$. The empirical Gaussian complexity of this class of linear classifiers can be estimated according to [1](Lemma 22) by

$$\frac{4 \cdot B \cdot (B+1) \cdot (B+2) \cdot \sqrt{m}}{m}.$$

The empirical Gaussian complexity and the Gaussian complexity differ by more than ϵ with probability at most $2 \cdot \exp(-\epsilon^2 m/8)$ according to [1](Theorem 11).

Since we can interprete the mapping $(x \mapsto (x_1^2, \ldots, x_n^2))$ as feature map of a kernel, an estimation of the Gaussian complexity for the considered quadratic functions is also possible: for $x \mapsto \sum \lambda_i^j x_i^2$ with $\|\lambda^j\| \leq 1$ we can estimate the empirical Gaussian complexity by

$$\frac{2 \cdot B^2 \cdot \sqrt{m}}{m}$$

because of [1](Lemma 22), using again the fact $\|x\| \leq B$.

Thus, the overall error bound

$$\begin{aligned} E_P(f) &\leq \hat{E}_m^L(f) + \frac{4K \cdot p(p-1)(2B(B+1)(B+2)+B^2)\ln m}{\rho \cdot \sqrt{m}} \\ &\quad + \left(1 + \frac{8K \cdot p(p-1) \cdot \ln m}{\rho}\right)\sqrt{\frac{\ln 4/\delta}{2m}} \\ &\leq \hat{E}_m^L(f) + \frac{\ln m}{\rho \cdot \sqrt{m}} \cdot \sqrt{\ln(1/\delta)} \cdot \mathrm{O}(Kp^2B^3) \end{aligned}$$

with probability of at least $1-\delta$ arises. This term limits the generalization error for all classifiers of the form (3) with adaptive metric if only two classes are dealt with and inputs and weights are restricted by B. Note that this bound is independent of the dimensionality n of the data. It scales inversely to the margin ρ, i.e. the larger the margin the better the generalization ability.

This bound indicates that LGRLVQ includes the objective of structural risk minimization during training because the terms $M_f(x,y)$ which characterize the margin are directly contained in the cost function of LGRLVQ. Naturally, only the extremal margin values need to be limited and thus a restriction of the respective update to extremal pairs of prototypes would suffice. Thus, this argument even proposes schemes for active data selection if a fixed and static pattern set is available for training to speed the algorithm and improve its convergence.

4 Discussion

We have extended GRLVQ and SRNG by local adaptation schemes which allow us to determine prototype-wise or class-wise relevance profiles of a given classification task. Apart from a greater flexibility, this feature offers further insight into the classification behavior and possible underlying semantical issues since it identifies data dimensions relevant for the particular region of the data space for the classification at hand. Remarkably, this further flexibility does not decrease the excellent generalization ability of these methods. We have derived explicit generalization bounds for these local variants which are competitive to the more simple case of global relevance factors. As in the global case, dimensionality independent large margin bounds result, and the margin occurs explicitely as nominator of the cost function optimized during training. This observation proposes interesting active learning schemes which can considerably reduce the training time by focusing on relevant training data according to the margin. This possibility is currently investigated by the authors in the context of biomedical applications.

References

[1] P.L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning and Research* 3:463-482, 2002.

[2] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[3] M.Biehl, A.Gosh, B.Hammer, Learning Vector Quantization: the dynamics of Winner-Takes-All algorithms, accepted for *Neurocomputing*.

[4] O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In: *Advances in Neural Information Processing Systems 2002*.

[5] V. Cherkassky, D. Gehring, and F. Mulier. Comparison of adaptive methods for function estimation from samples. *IEEE Transactions on Neural Networks*, 7:969-984, 1996.

[6] C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20 (1995), 1-20.

[7] K. Crammer, R. Gilad-Bachrach, A. Navot, and A. Tishby. Margin analysis of the LVQ algorithm. In: *Advances in Neural Information Processing Systems 2002*.

[8] R.N. Davé. Fuzzy shell-clustering and application to circle detection in digital images. *International Journal of General Systems* 16:343-355, 1990.

[9] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:773-781, 1989.

[10] T. van Gestel, J. A. K. Suykens, B. de Moor, and J. Vandewalle. Automatic relevance determination for least squares support vector machine classifiers. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks*, 13–18, 2001.

[11] Y. Grandvalet. Anisotropic noise injection for input variables relevance determination. *IEEE Transactions on Neural Networks*, 11(6):1201–1212, 2000.

[12] S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. Pattern Recognition Letters, 23:401–417, 2002.

[13] E.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In: *IEEE CDC*, pages 761-766, San Diego, California, 1979.

[14] B. Hammer and K. Gersmann. A note on the universal approximation capability of support vector machines. Neural Processing Letters 17: 43-53, 2003.

[15] B. Hammer, A. Rechtien, M. Strickert, T. Villmann, Rule extraction from self-organizing networks. In: J. R. Dorronsoro (ed.), *ICANN 2002*, Springer, 877-882, 2002.

[16] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters* 21(1): 21-44, 2005.

[17] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters* 21(2):109-120, 2005.

[18] B. Hammer, M. Strickert, and T. Villmann. Prototype recognition of splice sites. In: U.Seiffert, L.C.Jain, and P.Schweitzer (eds.), *Bioinformatics using Computational Intelligence Paradigms*, Springer, pages 25-55, 2005.

[19] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks* 15:1059-1068, 2002.

[20] D. Haussler. *Convolutional kernels for dicrete structures*. Technical Report UCSC-CRL-99-10, Computer Science Department, University of California at Santa Cruz, 1999.

[21] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–315. Springer, 1999.

[22] T. Jaakkola, M. Diekhans, and D. Haussler. A discrimitive framework for detecting remote protein homologies. *Journal of Computational Biology* 7(1-2): 95-114, 2000.

[23] S. Kaski and J. Sinkkonen. A topography-preserving latent variable model with learning metrics. In: N. Allinson, H. Yin, L. Allinson, and J. Slack, editors, *Advances in Self-Organizing Maps*, pages 224–229, Springer, 2001.

[24] S. Kaski. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks* 12:936-947, 2001.

[25] T. Kohonen. *Self-Organizing Maps*, 3rd edition, Springer, 2001.

[26] T. Kohonen. Learning vector quantization. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 537–540. MIT Press, 1995.

[27] T. Kohonen and P. Somervuo. How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9):945-952, 2002.

[28] T. Martinetz, S. Berkovich, and K. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE TNN* 4(4):558-569, 1993.

[29] T. Martinetz and K. Schulten. Topology representing networks. *IEEE Transactions on Neural Networks* 7(3):507-522, 1993.

[30] F. Mulier. *Statistical Analysis of Self-Organization*. Ph.D. Thesis, University of Minnesota, Minneapolis, 1994.

[31] M. Pregenzer, G. Pfurtscheller, and D. Flotzinger. Automated feature selection with distinction sensitive learning vector quantization. *Neurocomputing* 11:19-29, 1996.

[32] A.S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, MIT Press, 1995.

[33] A.S. Sato and K. Yamada. An analysis of convergence in generalized LVQ. In L. Niklasson, M. Bodén, and T. Ziemke (eds.) *ICANN'98*, pages 172-176, Springer, 1998.

[34] B. Schölkopf. *The kernel trick for distances*. Technical Report MSR-TR-2000-51. Microsoft Research, Redmond, WA, 2000.

[35] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[36] S. Seo and K. Obermeyer. Soft learning vector quantization. *Neural computation* 15:1589-1604, 2003.

[37] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research* 2: 67-93, 2001.

[38] M. Strickert, T. Bojer, and B. Hammer. Generalized relevance LVQ for time series. In: G.Dorffner, H.Bischof, K.Hornik (eds.), *Artificial Neural Networks - ICANN'2001*, Springer, pages 677-683, 2001.

[39] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B, 58, 267-288, 1996.

[40] V. Vapnik. Statistical Learning Theory. Wiley-Interscience, 1998.

[41] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications* 16(2):264-280, 1971.

[42] T. Villmann and B. Hammer. *Metric adaptation and relevance learning in learning vector quantization.* Technical Report, Reihe P, Heft 247, FB Mathematik/Informatik, Universität Osnabrück, 2003.

[43] T. Villmann, E. Merenyi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks* 16(3-4): 389-403, 2003.

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/~wjamroga/techreports/
ISSN: 1860-8477

The IfI Review Board